Semantic Data Cleansing in Genome Databases

Heiko Müller

Humboldt University Berlin
Unter den Linden 6
10099 Berlin
Germany
hmueller@informatik.hu-berlin.de

Abstract

The abundance of errors in genome databases is a well-known fact. Major problems are errors in genome annotation. Performing biological experiments to eliminate them is time consuming and expensive. As a viable alternative, we introduce novel data cleansing methods for (semi-) automatic detection and correction of erroneous entries. Using a simple example we show the applicability of this cleansing approach. Our approach forms a sound basis for the solution of many open questions such as the efficient identification of erroneous entries, the specification of the cleansing process, management of alternative solutions until the correct one is identified, and efficient management of dependencies to react on changes to base data and avoid outdated data.

1. Genome Data is Dirty

1.1 Problems in Genome Databases

Increasing interest in genome data has lead to the availability of a multitude of public available genome databases today[1]. By genome data we mean nucleic acid (DNA and RNA), amino acid (protein) sequence data, and their structural and functional classification (*annotation*). The process of assigning meaning to sequence data by identifying regions of interest and determine function for them is defined as genome annotation [MGMB+03]. Two of the most daunting problems within this process are the integrated access of multiple sources and the quality of the retrieved data. The former problem is common to all integrated databases and is regarded elsewhere [SL90]. The latter problem, on the other hand, has so far been studied only marginally in the context of genome data, despite of the importance of high data quality for ongoing genome research.

Errors in genome data can result in improper target selection for biological experiments or pharmaceutical research, in turn resulting in loss of money. Missing, incomplete or erroneous information hinders the automatic processing and analysis of data. This leads to a loss in confidence and a rise in effort and frustration for the biologist. Fuzzy or incomplete knowledge in conjunction with erroneous base data makes annotation a highly error prone process. In [Bor00] it is stated that an average of 70% accuracy in predicting functional and structural features must be considered a success!

Several studies show the existence of errors in genome databases, e.g. [Bre99, ITAE+03]. In [Bre99] the error rate is estimated to be over 8% by comparing analysis results of three independent research groups annotating the proteome of *Mycoplasma genitalium* and counting the number of discrepancies between them. In [ITAE+03] the authors generate a highly reliable set of annotations by carefully using automatic methods and experimental evidence. They compare their results with existing annotations and with the results of solely automatically performed annotations. For the original annotations only 63% of functional assignments within both datasets are in total agreement, while for the solely automatic annotations the precision is estimated to be 74% for the most reliable set of predictions.

1.2 Errors in Genome Data Production

Producing incorrect data is intrinsic to the current process of generating genome data. The main causes for poor data quality in genome databases are

- **Experimental errors** due to unnoticed experimental setup failure or systematic errors. These errors are hard to detect by observation because of the diminutiveness of samples.
- **Analysis errors** due to the absence of fixed rules and knowledge guiding the annotation process which leads to misinterpretations and incomplete or invalid information (*miss-annotation*).

[1] for a comprehensive listing see http://nar.oupjournals.org/cgi/content/full/31/1/1/DC1

- **Transformation errors** while performing transformations of information from one representation into another or one medium to another, e.g., data input or translation of DNA sequences into protein sequences.
- **Propagated errors**, when erroneous data is used for the generation of new data, e.g., within the process of functional annotation of proteins.
- **Stale data**, i.e., unnoticed changes to base data on which a data item depends and that falsify it. Genome databases often contain integrated or derived data. Changes to the original data often remain unnoticed hindering the update of the derived information, leading to stale (outdated) data.

Propagation of errors is considered to be the major problem in genome data production, because existing data is very often used within the production process. In [GADTO02] a dynamic probabilistic model for error propagation in data annotation is developed. The authors show that the iterative annotation approach leads to a systematic deterioration of database quality. Concluding, there is a great need for data cleansing in genome databases to prevent loss of money in pharmaceutical research due to decisions based on erroneous data and to avoid further source pollution by error propagation.

2. State of the Art

Data cleansing comprises the identification and removal of errors in existing data sets to enhance the overall data quality. In most of the existing work [ACG02, GFSSS01, HS95, LLL00, ME97, RH01, VVSKS01] the focus is on data transformation, enforcement of simple integrity constraints, and duplicate elimination. Also, most of the papers describe how to identify errors but leave it to the domain expert to choose the right method of correction. This is due to the domain dependence of this task. Existing cleansing approaches are mainly concerned with producing an unified and consistent data set, i.e., addressing primarily syntactical problems and ignore the semantic problem of verifying the correctness of the represented information. There also exists statistical approaches [BS01] intending to detect and eliminate errors using statistical methods and for filling-in missing values.

Most of the existing work covers the domains of address or publication databases. These domains benefit from a clear definition of the semantics of the concepts used. Also, there exists only a small set of well defined rules and constraints as well as standardized lookup tables usable to identify and correct certain data, e.g., cities with their ZIP-Code, country names, etc. See [MF03] for a detailed classification and comparison of state-of-the-art data cleansing methods.

In the area of genome data there has been little work regarding data cleansing. In [GZK01] the semi-automated cleaning of RNA alignment databases is described. Here, programs that search for inconsistencies in RNA alignments reduce the number of potential annotation errors. The correction or database update is performed manually. There is also work reported on complete re-annotation of genome data to verify and correct annotation errors, the so-called second-generation annotation. For an overview on re-annotation projects see [OK02]. A complete re-annotation has the disadvantage of being time consuming because entries that are not erroneous are re-annotated as well. Also, the process of re-annotation has to be performed each time there is an update to some of the base data used within the re-annotation process.

3. Cleansing of Genome Data

3.1 Genome Data

Genome data comprises genome sequences and their annotations. Each annotation can be determined by biological experiments. Despite being determined experimentally, annotations are also derived automatically from sequence data by means of results generated using standard or specific bioinformatics algorithms, possibly operating on additional data sources. The results are interpreted with expert knowledge in form of annotation rules. The need for automatic annotation arises because manually annotation cannot keep up any more with the huge amounts of sequence data produced every day. By assigning putative annotations the sequence data is available for further research as quickly as possible.

For each sequence *s* we define the annotation as a list of annotation values a_i, denoted by $A = <a_1, ..., a_n>$. The tuple (s, A) is called sequence annotation. Each annotation value is the output from applying an annotation function f_i to s, $f_i(s) = a_i$. The annotation function f_i in turn is composed of evidence functions h_{i_j} displaying certain features of the sequence using additional data sources q_{i_k},

$$f_i = h_{i_1} \circ h_{i_2} \circ ... \circ h_{i_m}(q_{i_1}, ..., q_{i_k}).$$

An example for such an evidence function is a sequence similarity search calculating a similarity value for a pair of sequences. The choice of evidence functions in f_i and the specification of their combination currently lie solely within the responsibility of a domain expert.

3.2 Semantic Cleansing of Genome Data

The common problems within genome data are format inconsistency, duplicates (synonyms), homonyms and syntax errors in textual annotation, as reported in [BB96]. They can be handled using the known cleansing approaches listed in [MF03].

We regard the problem of semantic errors in annotation, caused by analysis errors, error propagation, and stale data, as the most pressing problem hindering genome data quality. This conforms to the importance ranking of errors in genome annotation, the Transitive Annotation-Based Scale (TABS) [OK02], which describes seven major cases of errors in genome annotation and ranks them according to their effects on error propagation. In TABS semantically wrong annotation has the highest impact value while syntax errors having the lowest. Annotation error means that

$$f_i(s) = h_{i_1} \circ h_{i_2} \circ \ldots \circ h_{i_m}(q_{i_1}, \ldots, q_{i_k})(s) = a_i$$

does not reveal the same result as an experimental setup would do.

Semantic cleansing of genome data aims at generating for each given sequence s an annotation A describing as exactly as possible structural and functional characteristics, i.e., providing the same results as an experimental setup would do. The most reliable way to achieve this is to perform the biological experiments. This is also the most time consuming and expensive way and is therefore impracticable. Another solution is the complete re-annotation, which has the above-mentioned disadvantages. Therefore, we want to choose in advance the subset of sequences and annotations that are erroneous and then correct them individually.

3.2.1 Error Detection

There are two common methods for selecting sequences and their annotations as candidates for re-annotation. The first is to check biological integrity constraints on the given annotations. An integrity constraint c is a function associating with each sequence annotation *(s, A)* a Boolean value. The function *c(s, A)* returns ***true*** if the constraint is satisfied by the given sequence annotation, otherwise it returns ***false***. An example for such a constraint would be "The translation of mRNA always starts at the codon 'ATG'" (see Section 4). Constraint violating sequence annotations are the candidates for re-annotation. From our current point of knowledge, there are only few of these hard constraints that allow such a Boolean classification or erroneous annotations.

Another way for selection of erroneous annotations is to verify the correctness of the original performed annotation. Unfortunately, most of the existing genome databases contain only sequence data and their annotations omitting detailed information about the annotation process. This hampers reproduction and validation of the generated results. We therefore need to define a re-annotation function f_i' to verify the correctness of the annotation. Given such a function f_i' a Boolean function t is used to decide whether given annotation value a_i is correct. The function $t(s, a_i, f_i')$ returns ***true*** if $f_i'(s) = a_i$, otherwise it returns ***false***. A simple implementation would require re-annotation of all entries with the above-mentioned disadvantages. To reduce processing cost we need to implement efficient methods to identify erroneous entries without complete execution of f_i' by exploiting knowledge about the evidence functions used in the specification of f_i'.

3.2.2 Error Correction

The correction of erroneous entries is performed by re-annotation. Within the re-annotation process manipulating sequences or data in the additional sources used can be necessary. Often several different changes can yield in the same result and it has to be decided which is the correct one. As this is not always immediately possible the resulting alternative solutions have to be managed. The availability of additional or updated information may then allow choosing the correct solution. Evidence functions can be used to collect arguments for or against each of the alternative solutions. The resulting evidence values indicate the confidence in the correctness of values. These can then be used for decision support or to exclude alternatives in advance.

3.3 Annotation Lineage

We define the annotation lineage for an annotation value $f_i(s) = a_i$ according to [CW01] as union of the actual subset of items $q_{i_k}^*$ from each of the sources q_{i_k} used in f_i that contributed to the derivation of a_i. Annotation lineage comprises those items that contribute to the original annotation value or to the correction or verification of an annotation value during data cleansing.

Defining and managing annotation lineage enhances documentation of annotation and enables effortless identification of candidate annotations that have to be checked when data entries within their lineage are updated. Annotation lineage is also of importance for those cases where alternative corrections are managed. Upon identification of the correct alternative, the now incorrect values have to be deleted and with them all further annotations that are based on them have to be re-annotated.

4. Experiments

Using the MySQL load files for ENSEMBL database [HBBC+02] (Release 7.29) we installed a local copy of the relational database in our IBM DB2 database system and checked the biological constraint "All translations start with the codon 'ATG'". Error detection is done using a simple SQL query filtering those translations starting with a codon different from 'ATG'. Using protein sequences imported from the Oracle dump-file release of SWISS-PROT/TrEMBL [BBAB+03] (released July 15, 2002)[2] as additional data, we defined a re-annotation function which calculates the correct start codon using automatic processing.

For re-annotation of miss-annotated translations we first translated the upper end of the corresponding transcript into the according protein sequence. We then

[2] see ftp://ftp.ebi.ac.uk/pub/contrib/swissprot/oracle/README.html for more details

aligned protein sequences starting with the amino acid Methionine from SWISS-PROT/TrEMBL against the translated transcript. If such an alignment exists for the translated transcript the left end of this alignment marks the position of the new start codon. In some cases one of the sequences had to be modified to obtain an exact alignment.

About 30% of the translation entries in the ENSEMBL release violated this constraint. For nearly 15% of these violating entries a new start codon was proposed by our re-annotation function. A first survey of the ensuing releases of ENSEMBL and SWISS-PROT/TrEMBL showed that the database curators have also updated some of the identified corrections, enabling us to validate our methods.

5. Future Work and Conclusions

We defined semantic cleansing of genome data as the process of assuring correctness of annotations for genome sequences. This is performed by identifying erroneous annotations and re-annotating them. Using a simple example we validated the applicability of this approach and identified open problems and challenges for reliable cleansing of genome data.

Semantic cleansing of genome data is closely related to genome annotation. Both require domain dependent evidence functions. The definition of a set of general evidence functions for the domain of genome annotation will enable us to build a formal model to specify the annotation and cleansing process. Several additional challenges arise for the management of high quality genome data in database management systems. These challenges are metadata management for annotation rules, annotation lineage, and evidence values as well as management of alternative solutions (*versioning*).

The management of annotation rules and annotation lineage enable effective correctness verification. In those cases where the original annotation process and the data lineage are unknown the intrinsic properties of the evidence functions within the re-annotation specification can be used to detect erroneous annotations without the necessity of complete re-annotation. Including annotation lineage further enables efficient detection and re-annotation of affected annotations when changes in external data sources occur.

In those cases where alternative solutions and evidence values for them are managed it is desirable to include them within the annotation and cleansing process to receive results of higher quality. Some of the genome databases are also beginning to manage such evidences for their entries. Excluding invalid or unreliable entries from the processing can derive credible annotations. The formal model for genome annotation has to take evidence values and alternative solutions into account. Annotation lineage in conjunction with versioning enables identification of those items becoming invalid when alternative solutions are dismissed.

Acknowledgements

I thank my advisors Johann-Christoph Freytag and Bernhard Thalheim, as well as Felix Naumann. This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK~316).

References

[ACG02] R. Ananthakrishna, S. Chaudhuri, V. Ganti, *Eliminating Fuzzy Duplicates in Data Warehouses*, Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002

[BB96] P. Bork, A. Bairoch, *Go hunting in sequence databases but watch out for the* traps, Trends in Genetics, Vol 12, No. 10, 1996, 425-427

[BBAB+03] B. Boeckmann, A. Bairoch, R. Apweiler, M. Blatter, et al., *The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003*, Nucleic Acids Research Vol. 31, No. 1, 2003, 365-370

[Bor00] P. Bork, *Powers and Pitfalls in Sequence Analysis: The 70% Hurdle*, Genome Research, Vol. 10, No. 4 2000, 398-400

[Bre99] S.E. Brenner, *Errors in genome annotation*, Trends in Genetics, Vol. 15, No. 4, 1999, 132-133

[BS01] R. Bruni, A. Sassano, *Errors Detection and Correction in Large Scale Data Collecting*, in Advances of Intelligent Data Analysis, Lecture Notes in Computer Science 2189, Springer-Verlag, 2001

[CW01] Y. Cui, J. Widom, *Lineage Tracing for General Data Warehouse Transformations*, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[GADTO02] W.R. Gilks, B. Audit, D. DeAngelis, S. Tsoka, C. Ouzounis, *Modeling the prelocation of annotation errors in a database of protein sequences*, Bioinformatics, Vol. 18, No. 12, 2002, 1641-1649

[GFSSS01] H. Galhardas, D. Florescu, D. Shasha, E. Simon, C.-A. Saita, *Declarative data cleaning: Language, model, and algorithms*, Proceedings of the 27 th VLDB Conference, Roma, Italy, 2001

[GZK01] J. Gorodkin, C. Zwieb, B. Knudsen, *Semi-automated update and cleanup of structural RNA alignment databases*, Bioinformatics, Vol. 17, No. 7, 2001, 642-645

[HBBC+02] T. Hubbard, D. Barker, E. Birney, G. Cameron, et al., *The Ensembl genome database project*, Nucleic Acids Research, Vol. 30, No. 1, 2002, 38-41

[HS95] M.A. Hernandez, S.J. Stolfo, *The merge/purge problem for large databases*, Proceedings of the ACM SIGMOD Conference, San Jose, USA, 1995

[ITAE+03] I. Illopoulos, S. Tsoka, M.A. Andrade, A.J. Enright, et al., *Evaluation of annotation strategies using an entire genome sequence*, Bioinformatics, Vol. 19, No. 6, 2003, 717-726

[LLL00] Mong Li Lee, Tok Wang Ling, Wai Lup Low, *IntelliClean: A knowledge-based intelligent data cleaner*, Proceedings of the ACM SIGKDD, Boston, USA, 2000

[ME97] A.E. Monge, C.P. Elkan, *An efficient domain-independent algorithm for detecting approximately duplicate database tuples*, Workshop on Data Mining and Knowledge Discovery, Tucson, USA, 1997

[MF03] H. Müller, J.-C. Freytag, *Problems, Methods and Challenges in Comprehensive Data Cleansing*, Technical Report HUB-IB-164, Humboldt-Universität zu Berlin, Institut für Informatik, 2003.

[MGMB+03] F. Meyer, A. Goesmann, A.C. McHardy, D. Bartels, et al., *GenDB – an open source genome annotation system for prokaryote genomes*, Nucleic Acid Research, Vol. 31, No. 8, 2003, 2187-2195

[OK02] C.A. Ouzounis, P.D. Karp, *The past, present and future of genome-wide re-annotation*, Genome Biology, Vol. 3, No. 2, 2002, comment2001.1-2001.6

[RH01] V. Raman, J.M. Hellerstein, *Potter's Wheel: An Interactive Framework for Data Transformation and Cleaning*, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[SL90] A.P. Sheth, J.A. Larson, *Federated database systems for managing distributed, heterogeneous, and autonomous databases*, ACM Computing Surveys, Vol. 22, No. 3, 1990, 183-236

[VVSKS01] P. Vassiliadis, Z. Vagena, S. Skiadopoulos, N. Karayannidis, T. Sellis, *ARKTOS: towards the modeling, design, control and execution of ETL processes*, Information Systems, Vol. 26, 2001, 537-561